View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 55 (2015) 308 – 317

Procedia Computer Science

Information Technology and Quantitative Management (ITQM 2015)

A-TOPSIS – An approach Based on TOPSIS for Ranking Evolutionary Algorithms

Renato A. Krohling[a,b*], André G. C. Pacheco[b]

[a]Production Engineering Department, UFES - Federal University of Espirito Santo, Av. Fernando Ferrari 514, Vitória CEP: 29060-270, Brazil

[b]Graduate Program in Computer Science, PPGI, UFES - Federal University of Espirito Santo, Av. Fernando Ferrari 514, Vitória CEP: 29060-270, Brazil

Abstract

In this paper, we propose an alternative novel method based on the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) to solve the problem of ranking and comparing algorithms. In evolutionary computation, algorithms are executed several times and then a statistic in terms of mean values and standard deviations are calculated. In order to compare algorithms performance it is very common to handle such issue by means of statistical tests. Ranking algorithms, e.g., by means of Friedman test may also present limitations since they consider only the mean value and not the standard deviation of the results. Since the TOPSIS is not able to handle directly this kind of data, we develop an approach based on TOPSIS for algorithm ranking named as A-TOPSIS. In this case, the alternatives consist of the algorithms and the criteria are the benchmarks. The rating of the alternatives with respect to the criteria are expressed by means of a decision matrix in terms of mean values and standard deviations. A case study is used to illustrate the method for evolutionary algorithms. The simulation results show the feasibility of the A-TOPSIS to find out the ranking of algorithms under evaluation.

© 2015 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ITQM 2015

Keywords: Algorithms comparison, ranking, TOPSIS, evolutionary algorithms

1. Introduction

A great difficulty in evolutionary computation is the comparison of algorithms. Usually, the algorithms are run several times to multiple benchmarks. Then, the results are analyzed by means of statistical hypothesis tests [1-2]. The statistical tests can detect if there are differences between the performances of the algorithms. The problem is if there are differences, which algorithm is the best one? To use statistical tests in this step, it is necessary to make pairwise comparisons between the algorithms. Obviously, the number of tests required increases greatly with the number of algorithms being analyzed. This is problematic, first because the tiresome

* Corresponding author. Tel.: +55-27 4009-2649; fax: +55-27-4009 2649.
*E-mail address:*krohling.renato@gmail.com.

1877-0509 © 2015 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).
Peer-review under responsibility of the Organizing Committee of ITQM 2015
doi:10.1016/j.procs.2015.07.054

work of comparing each pair of algorithms; secondly and more important, the probability of making a mistake increases.

The *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS) developed by Hwang & Yoon [3] is a technique to evaluate the performance of alternatives through the similarity with the ideal solution. According to this technique, the best alternative would be one that is closest to the positive-ideal solution and farthest from the negative-ideal solution. The positive-ideal solution is one that maximizes the benefit criteria and minimizes the cost criteria. The negative-ideal solution maximizes the cost criteria and minimizes the benefit criteria. In summary, the positive-ideal solution is composed of all best values attainable of criteria, and the negative-ideal solution consists of all the worst values attainable of criteria. The interested reader shall refer to [4] for a broad survey about the TOPSIS.

TOPSIS has also been extended to treat, in a direct way, data expressed as probability distributions by means of the Hellinger distance [5]. It means that the TOPSIS with Hellinger distance [6] has opened a new possibility for ranking alternatives expressed in terms of probability distributions in the context of MCDM problems. Due to the stochastic nature of the evolutionary algorithms, in many cases the performance of evolutionary algorithms are expressed in terms of mean and standard deviation. It is not necessary to know the exact distribution of the solution of an algorithm. By the Central Limit Theorem, we know that $\bar{X} \xrightarrow{D} N(\mu, \frac{\sigma^2}{r})$. Therefore, if the algorithm is performed r times with r sufficiently large, we can approximate the distribution of the mean of the results by the Gaussian distribution and apply the Hellinger-TOPSIS to provide a rank order of the algorithms in a direct way. In the context of algorithms comparison, the alternatives consist of multiple algorithms and the criteria are the benchmarks.

In this paper, our goal is to modify our previous work [7] to handle a decision matrix with ratings evaluated in terms of mean and standard deviations aiming to present a tool to aid in selecting the best algorithms when applied to multiple benchmarks evaluated in terms of mean and standard deviations. The remainder of this article is organized as follows: Section 2 describes the TOPSIS. In Section 3, we present a novel approach based on TOPSIS to deal with a decision matrix with ratings in terms of means and standard deviations. In Section 4, simulation results for a case study concerning to dynamic optimization problems involving different versions of genetic algorithms applied to a suite of benchmarks problems are shown in order to illustrate the feasibility of the proposed approach. In Section 5, conclusions and directions for future work are given.

2. Multicriteria decision making

Let us consider the decision matrix D, which consists of a*lternatives* and *criteria,* described by:

$$D = \begin{array}{c} \\ A_1 \\ \ldots \\ A_m \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & \ldots & C_n \end{array} \\ \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \end{array} \tag{1}$$

where $A_1, A_2, \cdots, A_m$ are viable alternatives, and $C_1, C_2, \cdots, C_n$ are criteria, x_{ij} indicates the rating of the alternative A_i according to criteria C_j. The weight vector $W = (w_1, w_2 ..., w_n)$ is composed of the individual weights $w_j (j = 1,...,n)$ for each criterion C_j satisfying $\sum_{j=1}^{n} w_j = 1$. In general, the criteria are classified into two types: *benefit* and *cost*. The *benefit* criterion means that a higher value is better while for the *cost* criterion is valid the opposite. The data of the decision matrix D come from different sources, so it is necessary to normalize it in order to transform it into a dimensionless matrix, which allow the comparison of the various criteria. In this work, we use the normalized decision matrix $R = \left[r_{ij} \right]_{mxn}$ with $i = 1,...,m,$ and $j = 1,...,n.$ The normalized value r_{ij} is calculated as:

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{m} x_{ij}^2}}, \text{with } i = 1,...,m; j = 1,...,n \tag{2}$$

$$r_{ij} = \frac{x_{ij}}{x_{i\max}}, \text{with } i = 1,...,m; j = 1,...,n \tag{3}$$

The normalized decision matrix R represents the relative rating of the alternatives. After normalization, we calculate the weighted normalized decision matrix $P = \left[p_{ij} \right]_{mxn}$ with $i = 1,...,m,$ and $j = 1,...,n$ by multiplying the normalized decision matrix by its associated weights. The weighted normalized value p_{ij} is calculated as:

$$p_{ij} = w_i \cdot r_{ij} \text{ with } i = 1,...,m, \text{ and } j = 1,...,n \tag{4}$$

The Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) is described in the following steps [8]:

Step 1: Identify the positive ideal solutions A^+ (benefits) and negative ideal solutions A^- (costs) as follows:

$$A^+ = (p_1^+, p_2^+, ..., p_m^+) \tag{5}$$

$$A^- = (p_1^-, p_2^-, ..., p_m^-) \tag{6}$$

where

$$p_j^+ = \left(\max_i p_{ij}, j \in J_1; \min_i p_{ij}, j \in J_2 \right)$$

$$p_j^- = \left(\min_i p_{ij}, j \in J_1; \max_i p_{ij}, j \in J_2 \right)$$

where J_1 and J_2 represent the criteria *benefit* and *cost,* respectively.

Step 2: Calculate the Euclidean distances from the positive ideal solution A^+ (benefits) and the negative ideal solution A^- of each alternative A_i, respectively as follows:

$$d_i^+ = \sqrt{\sum_{j=1}^{n} (d_{ij}^+)^2} \tag{7}$$

$$d_i^- = \sqrt{\sum_{j=1}^{n} (d_{ij}^-)^2} \tag{8}$$

where

$d_{ij}^+ = p_j^+ - p_{ij}$, with $i = 1,...,m.$

$d_{ij}^- = p_j^- - p_{ij}$, with $i = 1,...,m.$

Step 3: Calculate the relative closeness ξ_i for each alternative A_i with respect to positive ideal solution as given by:

$$\xi_i = \frac{d_i^-}{d_i^+ + d_i^-} \tag{9}$$

Step 4: Rank the alternatives according to the relative closeness. The best alternatives are those that have higher value ξ_i and therefore should be chosen because they are closer to the positive ideal solution.

Next, we describe the TOPSIS approach involving two decision matrices described in terms of means and standard deviations.

3. A-TOPSIS - A novel approach based on TOPSIS for ranking algorithms

The decision matrix D consisting of a*lternatives* and *criteria* is described by

$$D = \begin{array}{c} \\ A_1 \\ ... \\ A_m \end{array} \begin{array}{c} \begin{array}{ccc} C_1 & ... & C_n \end{array} \\ \begin{pmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{pmatrix} \end{array} = \begin{pmatrix} (\mu_{11}, \sigma_{11}) & \cdots & (\mu_{1n}, \sigma_{1n}) \\ \vdots & \ddots & \vdots \\ (\mu_{m1}, \sigma_{m1}) & \cdots & (\mu_{mn}, \sigma_{mn}) \end{pmatrix}$$

where $A_1, A_2, \cdots, A_m$ are alternatives, $C_1, C_2, ..., C_n$ are criteria, $\tilde{x}_{ij}$ indicates the rating of the alternative A_i with respect to criterion C_j described in terms of its mean and standard deviations μ_{ij}, σ_{ij}, respectively. In the context of algorithms comparison, the alternatives consists of the algorithms and the criteria are the benchmark problems.

The problem consists of two decision matrices as given by $D = \{M_\mu, M_\sigma\}$.

$$M_\mu = \begin{pmatrix} \mu_{11} & \cdots & \mu_{1n} \\ \vdots & \ddots & \vdots \\ \mu_{m1} & \cdots & \mu_{mn} \end{pmatrix} \quad M_\sigma = \begin{pmatrix} \sigma_{11} & \cdots & \sigma_{1n} \\ \vdots & \ddots & \vdots \\ \sigma_{m1} & \cdots & \sigma_{mn} \end{pmatrix}$$

In this context, we develop a new framework combining the TOPSIS for ranking Evolutionary Algorithms in terms of mean values and standard deviations as illustrated in Fig. 1.

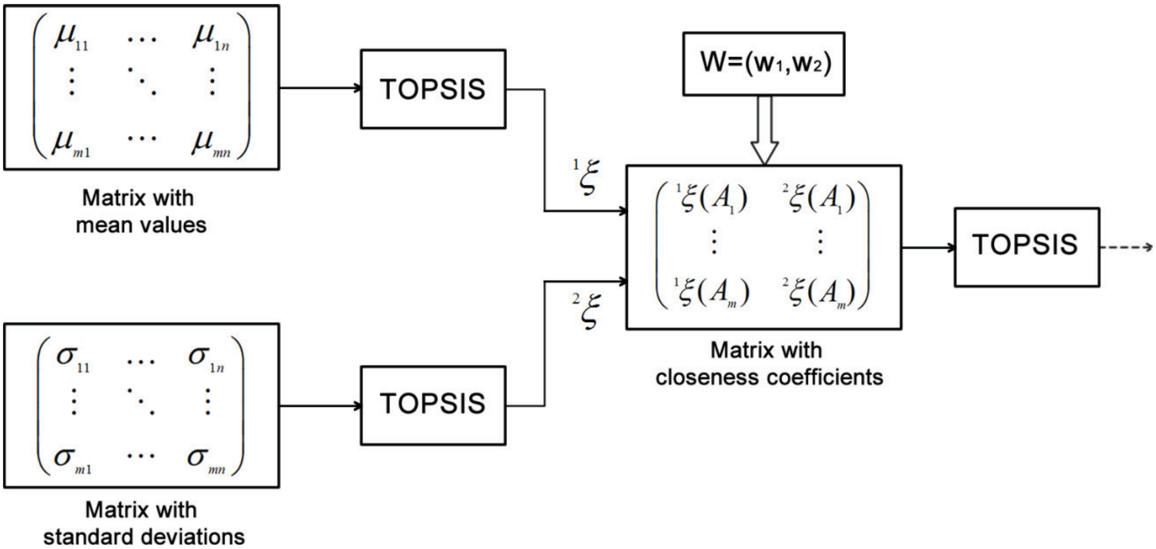

Fig. 1: Illustration of the A-TOPSIS approach for ranking evolutionary algorithms in terms of mean values and standard deviations.

The steps of the A-TOPSIS algorithm are described as follows:

Step 1: Normalize the matrices M_μ and M_σ.

Step 2: Identify the positive ideal solutions A^+ (benefits) and negative ideal solutions A^- (costs) for each matrix as follows:

$$A^+ = (p_1^+, p_2^+, ..., p_m^+) \tag{10}$$

$$A^- = (p_1^-, p_2^-, ..., p_m^-) \tag{11}$$

where

$$p_j^+ = \left(\max_i p_{ij}, j \in J_1; \min_i p_{ij}, j \in J_2 \right)$$

$$p_j^- = \left(\min_i p_{ij}, j \in J_1; \max_i p_{ij}, j \in J_2 \right)$$

where J_1 and J_2 represent the criteria *benefit* and *cost,* respectively.

Step 3: Calculate the Euclidean distances from the positive ideal solution A^+ (benefits) and the negative ideal solution A^- of each alternative A_i, respectively as follows:

$$d_i^+ = \sqrt{\sum_{j=1}^{n} (p_j^+ - p_{ij},)^2} \quad \text{with } i = 1, ..., m \tag{12}$$

$$d_i^- = \sqrt{\sum_{j=1}^{n} (p_j^- - p_{ij})^2} \quad \text{with } i = 1, ..., m \tag{13}$$

Step 4: Calculate the relative closeness for each alternative $\xi_i(A_i)$ with respect to positive ideal solution as:

$$\xi(A_i) = \frac{d_i^-}{d_i^+ + d_i^-} \quad \text{with } i = 1,...,m \tag{14}$$

Step 5: After calculating the vector $\xi_i(A_i)$ for both decision matrices we obtain a resulting decision matrix *G*, which is made up of the two vectors of the relative-closeness coefficients given by:

$$G = \begin{pmatrix} {}^1\xi(A_1) & {}^2\xi(A_1) \\ \vdots & \vdots \\ {}^1\xi(A_m) & {}^2\xi(A_m) \end{pmatrix} \tag{15}$$

In this case, to each of the vector is assigned a weight $W = (w_1, w_2) = (w_\mu, w_\sigma)$ where w_μ and w_σ represents the weight assigned to the criteria means, and standard deviations, respectively, which satisfies $w_\mu + w_\sigma = 1$. One can now obtain the weighted relative-closeness coefficient matrix by introducing the importance weights to each one of the relative-closeness coefficient vector as given by:

$$G = \begin{pmatrix} w_1\,{}^1\xi(A_1) & w_2\,{}^2\xi(A_1) \\ \vdots & \vdots \\ w_1\,{}^1\xi(A_m) & w_2\,{}^2\xi(A_m) \end{pmatrix} \tag{16}$$

From this stage on our method continues by applying the standard TOPSIS to the resulting matrix in order to identify the global ranking.

Step 6: Identify the global positive ideal solution A_G^+ and global negative ideal solution A_G^-, respectively as follows:

$$A_G^+ = (p_{G1}^+, p_{G2}^+) = \left(\max_i {}^l\xi(A_i),\ l \in J_1; \min_i {}^l\xi(A_i)\ l \in J_2 \right) \tag{17}$$

$$A_G^- = (p_{G1}^-, p_{G2}^-) = \left(\min_i {}^l\xi(A_i),\ l \in J_1; \max_i {}^l\xi(A_i)\ l \in J_2 \right) \tag{18}$$

where J_1 and J_2 represent the criteria *benefit* and *cost,* respectively.

Step 7: Calculate to each alternative A_i the distances from the global positive ideal solution A_G^+ and from the global negative ideal solution A_G^-, respectively as follows:

$$d_{Gi}^+ = \sqrt{\sum_{l=1}^{2} ({}^l\xi(A_i) - p_{Gl}^+)^2} \quad \text{with } i = 1,...,m \tag{19}$$

$$d_{Gi}^- = \sqrt{\sum_{l=1}^{2} ({}^l\xi(A_i) - p_{Gl}^-)^2} \quad \text{with } i = 1,...,m \tag{20}$$

Step 8: Calculate the global relative-closeness ξ_{Gi} for each alternative A_i with respect to global positive ideal solution A_G^+ as:

$$\xi_G(A_i) = \frac{d_{Gi}^-}{d_{Gi}^- + d_{Gi}^+} \tag{21}$$

Step 9: Rank the alternatives according to the relative closeness coefficients. The best alternatives are those that have higher value $\xi_G(A_i)$ and therefore should be chosen because they are closer to the positive ideal solution.

In order to compare our approach, we present another different way to calculate the final ranking using the geometric mean between the closeness coefficients [9], which are obtained by means of the application of TOPSIS to the decision matrix of means and standard deviations M_μ, M_σ, respectively as given by

$$\xi_G(A_i) = \sqrt{{}^1\xi(A_i) \cdot {}^2\xi(A_i)} \tag{22}$$

3. Simulation Results

Case study

Let us consider the optimization with evolutionary algorithms (or different versions originated from the same algorithm). There are in the literature established benchmark problems and without loss of generality we consider minimization problems, but maximization problems can be transformed easily to its equivalent minimization problems.

The G24 Benchmark Set of Dynamic Constrained Optimization Problems (DCOPs) was introduced by Nguyen [10] and Nguyen and Yao [11], which consists of a set of 18 benchmarks: 1) G24-u, 2) G24-1, 3) G24-f, 4) G24-uf, 5) G24-2, 6) G24-2u, 7) G24-3, 8) G24-3b, 9) G24-f, 10) G24-4, 11) G24-5, 12) G24-6a, 13) G24-6b, 14) G24-6c, 15) G24-6d, 16) G24-7, 17) G28-a, and 18) G24-8b.

The algorithms established in the literature for dynamic optimization problems are 21 different versions of Genetic Algorithms (GA) [10-11]: 1) GAnoElit, 2) RIGAnoElit, 3) HyperMnoElit, 4) GAelit, 5) RIGAelit, 6) HyperMelit, 7) GA+Repair, 8) GA+RepairwUPGwNR, 9) GA+RepairwUPGwRR, 10) GA+RepairwUPCwNRR, 11) dRepairGA, 12) dRepairRIGA, 13) dRepairHyperM, 14) dRepairGAOOR, 15) dRepairRIGAOOR, 16).dRepairHyperMOOR, 17) Genocop, 18) GenocopwUPGwNRR, 19) GenocopwUPGwRR, 20) GenocopwUPCwNRR, 21) dGenocop.

As a standard procedure in evolutionary computation, the 21 different Genetic Algorithms versions have been applied to the 18 dynamic constrained minimization benchmarks and the experiment is repeated for each algorithm 50 times [10-11]. So, a statistic in terms of mean and standard deviation (stdDev) is calculated as shown in Table 1. For this case in particular, we do not need to normalize the data since the matrices M_μ and M_σ are already in the range [0,1].

The problem now is to determine the best algorithms in terms of effectiveness among the 21 algorithms analyzed. Applying the A-TOPSIS to the data in Table 1 provides the results for the ranking of the algorithms, which is shown in Table 2. As we can notice, the best algorithm is A15, using A-TOPSIS with $W = (w_1, w_2) = (0.5, 0.5)$ assigning the same importance for the mean and standard deviation as compared with TOPSIS with geometric mean. Next, we carry out a sensitivity study by varying the weights w_1, w_2 for mean and standard deviation, respectively. It is interesting to note that when increasing the weight w_1 from 0.5 to $0.6 \to 0.7 \to 0.8 \to 0.9 \to 1.0$ and consequently diminishing the weight w_2 to

0.5 to $0.4 \rightarrow 0.3 \rightarrow 0.2 \rightarrow 0.1 \rightarrow 0$, respectively the best alternative change from A15 to A20 and remains stable. One notice that the first 5 alternatives change the order in the rank but A20, A17, A21, A16, and A15 are the five best.

Table 1. Decision matrix, where alternatives represent the 21 algorithms and the criteria represent 18 benchmarks of dynamic constrained minimization problems [10-11].

Algorithm	G24-u (dF, noC)		G24-1 (dF, fC)		G24-f (fF, fC)		G24-uf (fF, noC)		G24-2 (dF, fC)		G24-2u (dF, noC)		G24-3 (fF, dC)		G24-3b (dF, dC)		G24-3f (fF, fC)	
	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev
.GAnoElit	0.3610	0.0640	0.8030	0.0410	0.8020	0.0650	0.5460	0.0330	0.4290	0.0610	0.2060	0.0210	0.8840	0.0930	0.9970	0.0950	11.430	0.2250
.RIGAnoElit	0.2340	0.0190	0.6330	0.0460	0.6680	0.0790	0.4490	0.0490	0.3510	0.0600	0.1390	0.0320	0.7420	0.1000	0.8490	0.0930	0.7910	0.0850
.HyperMnoElit	0.2490	0.0340	0.4500	0.0940	0.2190	0.0850	0.1480	0.0510	0.3040	0.0250	0.1440	0.0260	0.6920	0.0940	0.5060	0.0180	0.2480	0.0710
.Gaelit	0.2140	0.0370	0.5870	0.0850	0.2270	0.0650	0.0950	0.0440	0.3290	0.0740	0.1030	0.0220	0.3840	0.0920	0.6370	0.1010	0.2410	0.0510
.RIGAelit	0.1310	0.0340	0.4010	0.0460	0.1710	0.1000	0.0860	0.0280	0.2830	0.0210	0.1100	0.0300	0.3400	0.0450	0.4720	0.0530	0.2030	0.0420
.HyperMelit	0.1730	0.0420	0.4750	0.0600	0.2240	0.0520	0.0930	0.0320	0.3760	0.0550	0.1110	0.0300	0.5610	0.1040	0.5110	0.1150	0.1420	0.0580
.GA+Repair	0.4680	0.0590	0.2260	0.0240	0.0410	0.0110	0.2180	0.0180	0.2810	0.0360	0.2940	0.0290	0.1560	0.0080	0.1710	0.0190	0.0250	0.0080
.GA+RepairwUPGwNR	0.6020	0.2110	0.6240	0.2020	0.2380	0.1590	0.8130	0.2240	0.4970	0.0660	0.5730	0.0920	0.2390	0.1280	0.4210	0.1420	0.0400	0.0470
.GA+RepairwUPGwRR	0.5770	0.0820	0.6200	0.1530	0.2380	0.1370	0.8070	0.2260	0.4550	0.1090	0.5500	0.1280	0.2000	0.0980	0.3970	0.1050	0.0380	0.0080
.GA+RepairwUPCwNRR	0.3060	0.0840	0.1040	0.0250	0.0410	0.0090	0.2180	0.0300	0.2020	0.0270	0.1980	0.0150	0.0380	0.0070	0.0750	0.0130	0.0250	0.0040
.dRepairGA	0.3620	0.0630	0.1010	0.0220	0.0420	0.0110	0.2190	0.0730	0.1980	0.0300	0.2010	0.0230	0.0340	0.0050	0.0790	0.0120	0.0250	0.0020
.dRepairRIGA	0.2540	0.0480	0.0820	0.0150	0.0280	0.0060	0.1940	0.0430	0.1620	0.0210	0.1870	0.0110	0.0290	0.0040	0.0580	0.0070	0.0140	0.0020
.dRepairHyperM	0.3190	0.0340	0.0930	0.0230	0.0450	0.0100	0.2180	0.0500	0.1710	0.0260	0.1960	0.0240	0.0270	0.0050	0.0710	0.0140	0.0250	0.0050
.dRepairGAOOR	0.1560	0.0180	0.1040	0.0250	0.0410	0.0100	0.2480	0.0800	0.1960	0.0350	0.0840	0.0300	0.0350	0.0070	0.0750	0.0120	0.0260	0.0040
.dRepairRIGAOOR	0.1520	0.0260	0.0780	0.0140	0.0290	0.0090	0.1510	0.0320	0.1710	0.0210	0.0820	0.0100	0.0290	0.0050	0.0590	0.0130	0.0140	0.0020
.dRepairHyperMOOR	0.1750	0.0400	0.0910	0.0160	0.0430	0.0100	0.2490	0.0720	0.1610	0.0290	0.0960	0.0220	0.0260	0.0050	0.0740	0.0140	0.0270	0.0060
.Genocop	0.1200	0.0280	0.0990	0.0340	0.0200	0.0080	0.0300	0.0080	0.1770	0.0310	0.1200	0.0280	0.0990	0.0340	0.0200	0.0080	0.0300	0.0080
.GenocopwUPGwNRR,	0.3020	0.1010	0.4940	0.1670	0.1090	0.0720	0.1700	0.0570	0.7010	0.1000	0.3020	0.1010	0.4940	0.1670	0.1090	0.0720	0.1700	0.0570
.GenocopwUPGwRR	0.4120	0.1950	0.7190	0.1850	0.1070	0.0830	0.1700	0.0530	0.6380	0.1780	0.4120	0.1950	0.7190	0.1850	0.1070	0.0830	0.1700	0.0530
.GenocopwUPCwNRR	0.1230	0.0290	0.1030	0.0240	0.0220	0.0140	0.0290	0.0160	0.1380	0.0300	0.1230	0.0290	0.1030	0.0240	0.0220	0.0140	0.0290	0.0160
.dGenocop	0.0910	0.0350	0.0850	0.0240	0.0210	0.0140	0.0300	0.0300	0.0990	0.0280	0.0600	0.0330	0.0280	0.0070	0.0680	0.0220	0.0050	0.0030

Algorithm	G24-4 (dF,dC)		G24-5 (dF, dC)		G24-6a (2DR, hard)		G24-uf (fF, noC)		G24-6b(1R)		G24-6c (2DR, easy)		G24-7 (fF, dC)		G24-8a (nC, ONISB)		G24-8b (fF,OICB)	
	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev	mean	stdDev
.GAnoElit	0.7180	0.0950	0.4650	0.0610	0.6300	0.0850	0.5410	0.0430	0.5750	0.0570	0.7300	0.0770	0.9130	0.1340	0.4790	0.0540	1.3270	0.1110
.RIGAnoElit	0.6120	0.0520	0.4010	0.0480	0.4550	0.0650	0.3800	0.0510	0.4120	0.0510	0.4610	0.0500	0.7130	0.0220	0.3920	0.0400	1.2160	0.0930
.HyperMnoElit	0.5620	0.0620	0.3470	0.0680	0.4370	0.0670	0.4390	0.0620	0.4020	0.0500	0.4810	0.0420	0.5440	0.0790	0.3920	0.0190	1.1780	0.0990
.Gaelit	0.6270	0.0450	0.3730	0.0310	0.8260	0.1540	0.5710	0.0710	0.5630	0.0620	0.6140	0.1080	0.5180	0.0950	0.4090	0.0270	1.3030	0.0830
.RIGAelit	0.4920	0.0710	0.2590	0.0310	0.4580	0.0500	0.4260	0.0450	0.4130	0.0400	0.4270	0.0280	0.4590	0.0570	0.4100	0.0190	1.0850	0.1110
.HyperMelit	0.4940	0.0390	0.2970	0.0470	0.5750	0.0740	0.4690	0.0740	0.5270	0.0390	0.5850	0.0570	0.4780	0.0720	0.4060	0.0460	1.0810	0.0840
.GA+Repair	0.2110	0.0150	0.2360	0.0240	0.4310	0.0740	0.4270	0.0480	0.3900	0.0380	0.3540	0.0380	0.1810	0.0170	0.4960	0.0320	0.3910	0.0680
.GA+RepairwUPGwNR	0.3390	0.1370	0.2860	0.0590	0.7440	0.4390	0.7080	0.1850	0.6200	0.1360	0.5160	0.2080	0.3510	0.1630	0.4480	0.1000	1.3270	0.1160
.GA+RepairwUPGwRR	0.3560	0.1010	0.2810	0.0840	0.4540	0.1540	0.6560	0.1510	0.6130	0.1030	0.5880	0.1730	0.3120	0.1360	0.4150	0.0920	1.1490	0.2090
.GA+RepairwUPCwNRR	0.1780	0.0180	0.1810	0.0230	0.4080	0.0580	0.3810	0.0480	0.3880	0.0370	0.3410	0.0290	0.1720	0.0250	0.4680	0.0530	0.4280	0.0860
.dRepairGA	0.1700	0.0260	0.1810	0.0320	0.4220	0.0590	0.3930	0.0380	0.3860	0.0450	0.3560	0.0370	0.1810	0.0430	0.4380	0.0310	0.4180	0.0470
.dRepairRIGA	0.1400	0.0280	0.1520	0.0170	0.3660	0.0330	0.3460	0.0280	0.3230	0.0370	0.3150	0.0290	0.1540	0.0310	0.4480	0.0200	0.3410	0.0530
.dRepairHyperM	0.0590	0.0100	0.1310	0.0190	0.3580	0.0490	0.3410	0.0390	0.3260	0.0470	0.2860	0.0350	0.0670	0.0140	0.4130	0.0320	0.2570	0.0420
.dRepairGAOOR	0.1640	0.0310	0.1770	0.0340	0.3950	0.0480	0.3910	0.0450	0.3860	0.0370	0.3520	0.0350	0.1790	0.0470	0.4220	0.0370	0.4490	0.0750
.dRepairRIGAOOR	0.1430	0.0240	0.1540	0.0280	0.3610	0.0510	0.3520	0.0350	0.3500	0.0320	0.3020	0.0220	0.1530	0.0340	0.4490	0.0170	0.3390	0.0510
.dRepairHyperMOOR	0.0620	0.0110	0.1310	0.0190	0.3390	0.0380	0.3420	0.0400	0.3300	0.0340	0.2810	0.0360	0.0680	0.0150	0.3970	0.0380	0.2420	0.0380
.Genocop	0.1770	0.0310	0.0590	0.0390	0.0410	0.0090	0.4070	0.0730	0.2960	0.0500	0.2810	0.0500	0.2300	0.0520	0.4080	0.0430	0.4460	0.0950
.GenocopwUPGwNRR,	0.7010	0.1000	0.3780	0.1210	0.2930	0.1050	0.8090	0.1750	0.5570	0.0760	0.7250	0.3970	0.2570	0.0920	0.6820	0.1410	1.2730	0.1600
.GenocopwUPGwRR	0.6380	0.1780	0.4400	0.2790	0.2940	0.1160	0.6880	0.1870	0.5930	0.1760	0.5780	0.1890	0.4400	0.1230	0.7840	0.0930	1.3560	0.1930
.GenocopwUPCwNRR	0.1380	0.0300	0.0740	0.0250	0.0360	0.0080	0.3190	0.0740	0.2800	0.0490	0.2910	0.0610	0.1710	0.0330	0.4270	0.0390	0.4470	0.1010
.dGenocop	0.1400	0.0430	0.1140	0.0250	0.3150	0.0630	0.3340	0.0850	0.2630	0.0420	0.2420	0.0410	0.1920	0.0540	0.4150	0.0390	0.4160	0.0850

Table 2. Results of the rank for the 21 algorithms obtained by the proposed A-TOPSIS approach compared to that with TOPSIS using geometric mean [9].

A-TOPSIS with $(w_1, w_2) = (0.5, 0.5)$			A-TOPSIS with $(w_1, w_2) = (1, 0)$			TOPSIS using geometric mean		
Ranking	Algorithm	ξ_i	**Ranking**	Algorithm	ξ_i	**Ranking**	Algorithm	ξ_i
1	A15	0.9531	**1**	A20	1.0000	**1**	A15	0.9026
2	A16	0.9522	**2**	A17	0.9767	**2**	A20	0.9021
3	A20	0.9510	**3**	A21	0.9630	**3**	A16	0.8947
4	A17	0.9444	**4**	A16	0.9470	**4**	A12	0.8935
5	A13	0.9363	**5**	A15	0.9386	**5**	A17	0.8925
6	A12	0.9357	**6**	A13	0.9186	**6**	A13	0.8898
7	A21	0.9269	**7**	A12	0.9156	**7**	A21	0.8809
8	A14	0.8931	**8**	A14	0.8819	**8**	A14	0.8574
9	A10	0.8800	**9**	A10	0.8624	**9**	A10	0.8501
10	A11	0.8737	**10**	A11	0.8519	**10**	A11	0.8470
11	A7	0.8317	**11**	A7	0.7848	**11**	A7	0.8272
12	A5	0.6442	**12**	A5	0.5526	**12**	A5	0.7017
13	A6	0.5701	**13**	A6	0.4766	**13**	A6	0.6493
14	A3	0.5528	**14**	A3	0.4447	**14**	A3	0.6367
15	A4	0.4774	**15**	A18	0.4411	**15**	A4	0.5843
16	A2	0.4353	**16**	A9	0.4178	**16**	A2	0.5262
17	A9	0.3566	**17**	A19	0.3830	**17**	A9	0.5049
18	A18	0.3535	**18**	A4	0.3802	**18**	A18	0.4968
19	A1	0.2944	**19**	A8	0.3600	**19**	A19	0.4528
20	A19	0.2941	**20**	A2	0.2066	**20**	A8	0.4074
21	A8	0.2570	**21**	A1	0	**21**	A1	0.3747

5. Concluding remarks

In this work, we present the A-TOPSIS to compare performance among algorithms in terms of mean values and standard deviations. This method allows finding the best algorithm, the second better and the worst. In order to illustrate the method a realistic case involving benchmarks of constrained dynamic optimization is presented. The results show the effectiveness of the method. In terms of computational burden, the A-TOPSIS consists of a very simple computation procedure, which shall encourage researcher/practitioner in different areas of knowledge to use it. It is important to note that the TOPSIS is a well-stablished and reliable method.

Acknowledgments

Renato A. Krohling thanks the financial support of the of the Brazilian agency CNPq (grant number 303577/2012-6).

References

[1] Derrac, J., Garcia, S., Molina, D., Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 2011;**1:**3-18.

[2] García, S., Molina, D., Lozano, M., Herrera, F. A study on the use of nonparametric tests for analyzing the evolutionary algorithms' behaviour: A case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics* 2009; **15**:617-644.

[3] Hwang, C.L., Yoon, K.P. *Multiple attributes decision making methods and applications*. Berlin: Springer-Verlag; 1981.

[4] Behzadian, M., Otaghsara, S. K., Yazdani, M., Ignatius, J. *A state-of the-art survey of TOPSIS applications*. *Expert Systems with Applications* 2012;**39**:13051-13069.

[5] Hellinger, E. Neue Begründung der Theorie quadratischer Formen von unendlichvielen Veränderlichen. *Journal für die reine und angewandte Mathematik* 1909;**136**:210-271.

[6] Lourenzutti, R., Krohling, R.A. The Hellinger distance in multicriteria decision making: An illustration to the TOPSIS and TODIM methods. *Expert Systems with Applications* 2014;**41**:4414-4421.

[7] Krohling, R.A., Campanharo, V.C. Fuzzy TOPSIS for group decision making: A case study for accidents with oil spill in the sea. *Expert Systems with Applications* 2011;**38**:4190-4197.

[8] Huang, J. (2008). Combining entropy weight and TOPSIS method for information system selection. Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, CIS 2008, p. 1281-1284.

[9] Ye, F., Li, Y. An extended TOPSIS model based on the possibility theory under fuzzy environment. *Knowledge-Based Systems* 2014;**67**:263-269.

[10] Nguyen, T., Continuous dynamic optimisation using evolutionary algorithms. Ph.D. thesis, The University of Birmingham, Birmingham, UK. [Online]. Available: http://etheses.bham.ac.uk/1296; 2010.

[11] Nguyen, T., Yao, X. Continuous dynamic constrained optimization the challenges. *IEEE Transactions on Evolutionary Computation* 2012;**16**:769-786.